CHAPTER

13

THE PRINTING SYSTEM

After reading this chapter and completing the exercises, you will be able to:

- Explain how a Linux printing system operates
- Set up printing using standard utilities
- Manage printing on a busy Linux system
- Configure printing for remote computer systems

In the previous chapter you learned about creating shell scripts using `if/then` statements, `for` loops, and other commands. You learned how to execute these scripts from a command line and how to schedule a script or any command for later execution using the `at` and `crontab` commands.

In this chapter you will learn how to set up and manage printing on a Linux system. You will learn how printing occurs on Linux and what utilities are available for you to manage files printed by users. You will also learn how to set up and print to remote computers.

UNDERSTANDING LINUX PRINTING

The Linux printing system (which is usually referred to as LPRng) was originally designed to be similar to the one used in the BSD version of UNIX. For this reason, you'll sometimes hear that Linux uses a standard BSD printing system. This system allows many users to print files at the same time without interfering with one another. It also allows a system administrator to manage and track all printing activity. The next two sections explain the components and basic printing process in the Linux system.

The Printing Architecture

The printing architecture of Linux includes the following components. Each of these components is also shown in Figure 13-1, which illustrates how a file is printed. This process is described in detail following the list of components.

- `lpr`: a program that prepares files to be sent to a printer.
- **Print job**: a file submitted for printing via the `lpr` command.
- **Print filter**: a script used by the `lpr` program to prepare files to be sent to a printer.
- `lpd`: a daemon that sends files (previously prepared by `lpr`) to the printer.
- `printcap` (for *printer capture*): a printer definition file that specifies how and where files are stored and processed by `lpr` and `lpd`.
- **Print spool directory** (or print queue): directory where files are placed before being sent to the printer.
- Administrative utilities: programs used by a system administrator to manage files that have been printed or are in the process of being printed, the printer definition file (`printcap`), and the status of each printer. (These utilities are described in detail later in this chapter.)

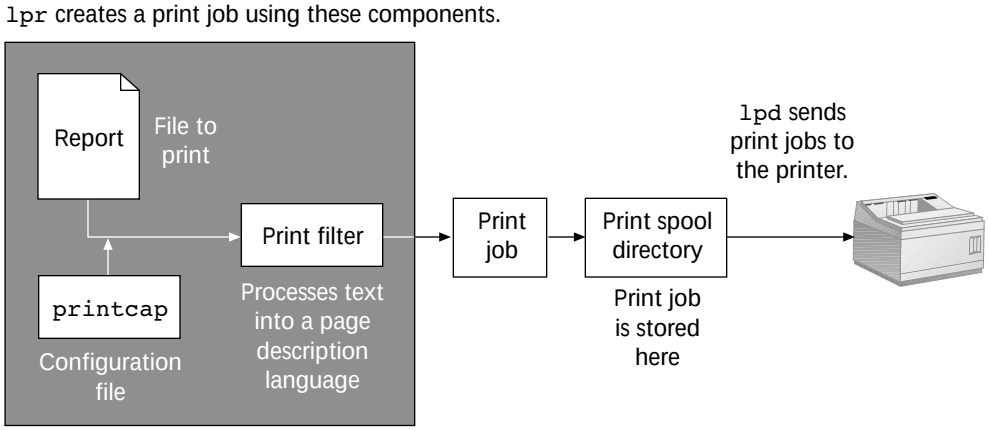

Figure 13-1 Printing a file in Linux

To print a file, you submit it to a print queue (or print spool directory) using the `lpr` command. Note that the terms *print spool directory* and **print queue** are used interchangeably to refer to a directory that stores files in the process of being printed. The `lpd` daemon retrieves files from the print spool directory one by one and sends them to the printer (for example, via the computer's parallel port).

A spool directory is a directory in which files are placed to await processing by another program. The print spool directory is only one of several spool directories within Linux.

The print spool directory is normally located within the `/var/spool/lpd` subdirectory. The print queue itself is often named for a printer. For example, you might define a print queue named `hplj` to store files that are destined to be printed on a Hewlett-Packard laserjet printer. If you define a print queue named `hplj`, the print queue directory, or print spool directory, would normally be `/var/spool/lpd/hplj`. Note, however, that the correlation between a print queue and a printer is not always one to one. That is, one print queue might be used to send print jobs to multiple printers. Conversely, several print queues might all specify the same printer.

To understand how this works, imagine a large network in which many users need to print files. All of the files might be submitted (by the `lpr` program on each networked computer) to a single print queue. The lines in the `printcap` file that define this single print queue might then list several printer devices (for example, the printers connected to parallel ports one and two). It is then the task of the `lpd` daemon to send each print job in order to maximize efficiency of the system (getting print jobs completed as rapidly as possible). Such an arrangement (shown in Figure 13-2) allows twice as many print jobs to be printed at the same time, without requiring that users decide which of two different print queues will be faster.

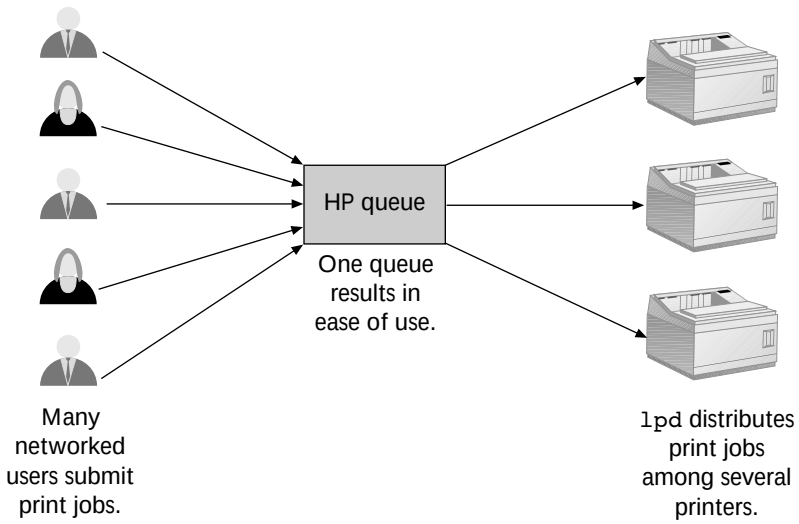

Figure 13-2 A single print queue sending files to many printers

A small network might employ a different setup. For example, a system administrator might configure separate print queues for envelopes, color printing, legal-sized documents, and standard printing. Users on the network would submit files for printing depending on the type of document being printed. The person maintaining the printer would then activate the

print queue for legal-sized documents (and disable all other print queues) only when legal-sized paper was inserted and ready to print. When envelopes were inserted in the single printer, the envelope queue could be activated, and so forth. This method requires that someone actively maintain the printer; it also requires users to wait while each queue is processed in turn. Figure 13-3 shows this arrangement graphically.

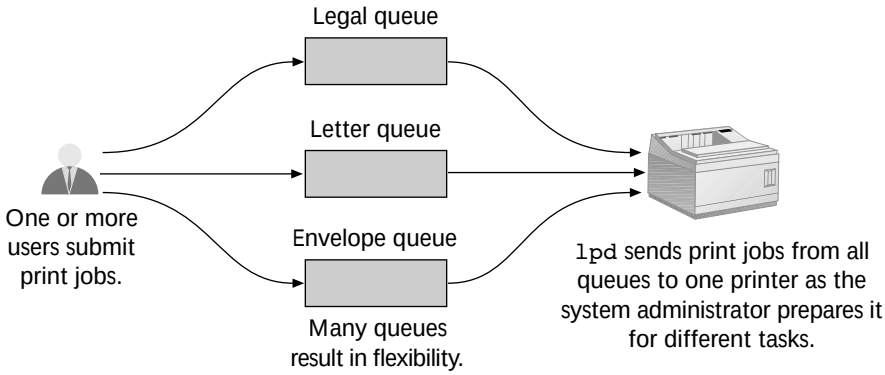

Figure 13-3 Many print queues sending to a single printer

Each print queue is configured by a series of lines in the `printcap` file. The information in this file that relates to a specific print queue is often called the printer definition. The `printcap` file, which is stored in the `/etc` directory, requires a complex format and can include dozens of options for each print queue that you define. For this reason, the `printcap` file is rarely configured manually (in a text editor). Instead, you can use one of the graphical tools (described later in this chapter) to set up the `printcap` file.

Although the `printcap` file defines print queues, system administrators refer to each print queue definition as a printer. When you refer to a printer (by a name such as `hplj`) in a Linux environment, you are actually referring to a print queue defined in the `printcap` file.

Print jobs are stored in the subdirectory assigned to a particular print queue. The `lpd` daemon watches for print jobs and then sends them one at a time to the physical printer device defined for that print queue. The `lpd` daemon also controls the actions allowed by the `lpr` command. For example, you could request that `lpd` disable a print queue so that users cannot submit new print jobs while you troubleshoot a printing problem.

The physical printer device is normally connected to a parallel or serial port. When you define the print queue, you specify which printer the files in that queue should be sent to; you do so by providing the name of the appropriate Linux device. For example, if the printer is attached to the first parallel port of the Linux computer, the device name is `/dev/lp0`. If the printer is attached to the third serial port, the device name is `/dev/ttyS2`. As you have learned in several previous chapters, these and other devices on Linux are accessed by referring to a filename in the `/dev` subdirectory. Device names accessed as filenames are sometimes called block special files or character special files. A **block special file** is a type of file (normally located in `/dev`) that refers to a physical device that transfers data in blocks of characters, such as a hard

disk drive. A **character special file** is a filename that refers to a physical device that transfers data in single characters, such as a serial port.

The output of the `ls -l` command includes an indicator of whether a filename refers to a block or character special file. The leftmost column displays a `b` or a `c`, for block or character special file, respectively. For example, the command `ls -l /dev/hda1` produces the following output. (Note the `b` in the left column.)

```
brw-rw----    1 root      disk       3,    1 May  5  1998 /dev/hda1
```

The command `ls -l /dev/ttyS0` creates this output. (Note the `c` in the left column, indicating a character special file.)

```
crw-------    1 root      tty        4,   64 May  5  1998 /dev/ttyS0
```

The Role of Print Filters

In order to print different types of documents from Linux applications to a variety of printer devices, Linux uses special programs that prepare data for printing. You have complete control over how these programs operate. To understand these programs, you first must understand something about how printers process the data that is sent to them.

Printer Languages

Each printer uses a page description language to guide what it prints. A **page description language** is a special set of codes that determine the graphics elements, text font, and everything else about what appears on a printed page. The most widely used page description languages are called PostScript and Printer Control Language, or PCL.

The content of the document you want to print and the format of the page must be converted into instructions that match the page description language used by the printer. Each type of printer requires a different set of steps to convert a document into the correct page description language codes.

The PostScript page description language was developed by Adobe and is widely used in many types of printers. Printers that use the PostScript codes are expensive relative to non-PostScript printers. PCL, developed by Hewlett-Packard, is used by most Hewlett-Packard printers, although these printers often provide support for PostScript as well. When you send a print job to a PostScript printer, the print filter (described in detail in the next section) creates a PostScript document, which consists of special instructions that tell the printer what to print. The following shows an example of the first part of a PostScript document:

```
%!PS-Adobe-3.0
%%Creator: groff version 1.05
%%DocumentNeededResources: font Times-Bold
%%+ font Times-Italic
%%+ font Times-Roman
%%DocumentSuppliedResources: procset grops 1.05 0
%%Pages: 5
%%PageOrder: Ascend
%%Orientation: Portrait
```

```
%%EndComments
%%BeginProlog
%%BeginResource: procset grops 1.05 0

/setpacking where {
      pop
      currentpacking
      true setpacking
} if

/grops 120 dict dup begin

% The ASCII code of the space character.
/SC 32 def

/A /show load def
/B { 0 SC 3 -1 roll widthshow } bind def
/C { 0 exch ashow } bind def
/D { 0 exch 0 SC 5 2 roll awidthshow } bind def
/E { 0 rmoveto show } bind def
/F { 0 rmoveto 0 SC 3 -1 roll widthshow } bind def
/G { 0 rmoveto 0 exch ashow } bind def
/H { 0 rmoveto 0 exch 0 SC 5 2 roll awidthshow } bind def
/I { 0 exch rmoveto show } bind def
/J { 0 exch rmoveto 0 SC 3 -1 roll widthshow } bind def
/K { 0 exch rmoveto 0 exch ashow } bind def
/L { 0 exch rmoveto 0 exch 0 SC 5 2 roll awidthshow } bind def
/M { rmoveto show } bind def
/N { rmoveto 0 SC 3 -1 roll widthshow } bind def
/O { rmoveto 0 exch ashow } bind def
/P { rmoveto 0 exch 0 SC 5 2 roll awidthshow } bind def
/Q { moveto show } bind def
/R { moveto 0 SC 3 -1 roll widthshow } bind def
/S { moveto 0 exch ashow } bind def
/T { moveto 0 exch 0 SC 5 2 roll awidthshow } bind def

% name size font SF -

/SF {
      findfont exch
      [ exch dup 0 exch 0 exch neg 0 0 ] makefont
      dup setfont
      [ exch /setfont cvx ] cvx bind def
} bind def

% name a c d font MF -

/MF {
      findfont
      [ 5 2 roll
      0 3 1 roll % b
```

```
      neg 0 0 ] makefont
      dup setfont
      [ exch /setfont cvx ] cvx bind def
} bind def

/level0 0 def
/RES 0 def
/PL 0 def
/LS 0 def
```

PostScript instructions could be represented in English like this: "Start a new page. Go to this position on the page, draw a short line, then a small circle. Go to this position, draw a small graphic image." Taken as a whole, these instructions result in letters, images, and other parts of a printed page.

Page description languages are very complex. Some printers can interpret only basic instructions and as a result provide limited functionality. For example, the printer may simply print the characters that are sent to it, having only the ability to interpret a few special characters that mark the end of a line, the end of a page, or other special situations.

Drivers and Filters

The process by which a print filter converts a document into the appropriate page description language codes (such as PostScript codes) is comparable to what happens on a Windows-based computer when you print a document. However, the Windows-based computer uses a different system to accomplish the same work. On Windows-based computers, the operating system includes hundreds of drivers for different printers. A **driver** is a small software program that provides abstract services for a hardware component. For example, a hard disk driver allows the operating system to simply say "open this file." The driver contains the instructions for locating a file on a specific type of hardware device. In the same way, a printer driver contains instructions that convert data into the format needed for a specific printer.

Unfortunately, Linux does not use printer drivers because this method—which requires close cooperation between the operating system vendor and printer manufacturer—was not popular when UNIX was first developed. Linux inherited the UNIX printing model, which allows flexibility, but requires more effort by users (using print filters) to convert documents into the correct printer format.

A few commercial programs such as Corel WordPerfect for Linux and, to a lesser extent, StarOffice for Linux (from Sun Microsystems) include hundreds of printer drivers that let you create complex documents using all the advanced features of a printer. These printer drivers are not available to other Linux programs, however, because Linux has no standard system for interacting with the printer drivers, even if you have the drivers installed on your system as part of a program such as WordPerfect for Linux.

Instead of using printer drivers, Linux uses print filters. A print filter is a script that contains instructions for formatting all documents for a specific printer using the page description language that the printer requires. The concept of a print filter is the same as a printer driver; the

difference is that a print filter is rarely as powerful. It does not provide access to advanced features of the printer. Print filters are also much more limited in the printer formats that they support. Print filters in Linux normally rely on a program that converts a text file into a page description language, including special printer codes for color printing, ejecting pages, and similar special functions. Examples of print filter programs include the following:

- `gs (GhostScript)`
- `enscript`
- `nenscript`

Each of these programs lets you provide the name of a printer device as a parameter. (This is done within the print filter.) The program then outputs data that `lpd` sends directly to the printer device. The output contains the correct page description language codes for the printer. Below you'll find a simple example of a print filter; this example illustrates how most print filters operate. A variable at the top of the script indicates the type of printer that the filter prepares data for. That variable is used later in the script when the `gs` or `enscript` program is executed. The output is stored in a file so that the `lpd` program can send the file directly to the printer.

```
#!/bin/bash
source /etc/sysconfig/printers/lp
if [ "$PAPERSIZE" = "a4" ]; then
  T=A4
else
  T=Letter
fi

enscript -M $T -Z -p - |

if [ "$DOUBLEPAGE" = "true" ]; then
  psnup -d -b0.6cm -p$PAPERSIZE -2
else
  cat -
fi |

if [ "$GSDEVICE" = "PostScript" ]; then
      cat -
elif [ "$GSDEVICE" = "uniprint" ]; then
      exec 3>&1 1>&2
      gs @$UPP.upp -q -sOutputFile="|cat 1>&3"
else
      gs -q $GSOPTIONS -sDEVICE=$GSDEVICE \
            -r$RESOLUTION \
            -sPAPERSIZE=$PAPERSIZE \
            -dNOPAUSE \
            -dSAFER \
            -sOutputFile=- -
fi
```

```
if [ "$SENDEOF" != "" ]; then
      printf "\004"
fi

exit 0
```

The print filter script is normally stored in the print queue subdirectory, along with the print jobs themselves. A separate print filter is used for each print queue. For example, if you defined a printer named `hplj`, the filter that processed documents going to that print queue would probably be `/var/spool/lpd/hplj/filter`. Because each print filter is located in the subdirectory of a specific print queue, all print filters are normally named `filter` or something similar. The `printcap` file specifies the location and script name for each print queue that is defined.

Normally your Linux distribution will be set up to use a magic filter. A **magic filter** is a filter program that automatically processes a file into the correct output format based on the file's type. For example, graphics files will be processed differently than plain text files. The most widely used magic filters that you will see mentioned in the `printcap` file or within a filter script file are called `APSfilter` and `magicfilter`.

SETTING UP PRINTING

When you install most Linux systems, the printing system is usually configured and running, except for the `printcap` file. (On some Linux systems, the printer is also configured as you install the operating system, so everything is ready for you to print files.) The Red Hat Linux installation allows you to configure a printer (create a `printcap` file). Note that the installation steps in Chapter 3 do not cover configuring a printer. Later in this chapter, you will learn how to use the Printer Tool program to configure a printer in Red Hat Linux. In general, the print configuration that you can perform during an installation is very basic. This part of the chapter describes how to define print policies and use various utilities to set up a robust printing system on a Linux-based computer.

The following list summarizes the steps required to get a printer working on Linux. These steps discuss editing a print configuration file in a text editor because this method works on all Linux distributions. In practice, however, you will almost always use a graphical utility specific to the Linux distribution you are using. (The Red Hat Printer Tool program described later is one example of this type of utility.) These steps are described in more detail in the following sections.

1. Connect a printer to the Linux computer using a parallel or serial cable.
2. Check the initialization scripts in the `/etc/rc.d/init.d` directory and the run level directories (such as `/etc/rc.d/rc3.d`) to be certain that the `lpd` daemon is started at boot time. Then use the `ps` command to be certain that the `lpd` daemon is in fact running.
3. Add an entry to the `printcap` file for the printer you have connected to the system.

4. Create the spool directory, filter, and accounting files, as appropriate. (This is normally done by the graphical utility that you use to configure the `printcap` file.)
5. Start the printer using the `lpc` command. (This is normally done by the graphical utility that you use to configure the `printcap` file.)
6. Test the printer by printing a small file.

Deciding on Print Policies

When overseeing a Linux system with many user accounts, the system administrator is likely to spend a great deal of time managing printing. Many of the other management tasks are either one-time tasks (such as creating user accounts or installing new software packages) or are necessary only when problems arise (and thus can be largely eliminated on a well-planned system). Printing, on the other hand, involves hardware (the printer) that is more subject to breakdown than almost anything inside a computer. Printing consumes resources such as paper and toner that must be replenished even when the system is working correctly. Finally, printers are a resource that is shared among all users, just as hard disk space is. What's more, printers are inherently hard to manage because they are slower, more expensive, and much more limited in capacity than hard disks.

To illustrate the differences between managing a hard disk used as a shared file server and a printer used by many users, consider these points:

- A single large hard disk will meet the needs of several dozen users on a standard system. Regular data backups and a redundant system such as disk mirroring provide protection against problems, but once users know where their data is stored and file permissions have been established, the system will generally run a long time without serious problems.
- Several dozen users probably require several printer devices, depending on the type of work they do. Each printer may cost as much as the single large hard disk. Supplies for the printers are a continuing cost.
- For users who print numerous reports, artwork sheets, or mechanical drawings, the printers become a bottleneck. If many users submit large print jobs at the same time, they must wait for the printer to finish each print job. Conversely, when a user saves data to a hard disk, the wait is never more than a second or two.
- The hard disk on which users store data is probably located in a locked closet. Only the system administrator has access to it. The printers, by contrast, are located where all users have access to them. Any user may try to adjust the printer features, cause a paper jam, or do other unexpected things. The actions of one user can make the printer unusable until the system administrator physically goes to the printer to solve the problem.

For all of these reasons, a printer policy is a helpful document for any organization with more than two or three users who rely on the same printer. A **printer policy** is a brief document that describes how print resources can be used and how the management of the printers will be conducted. Having a printer policy that everyone in the organization can review will help

a system administrator avoid the headaches that come with multiple hardware failures and demands from users for special treatment. A typical printer policy would include statements about the following issues. (You will learn later in this chapter how to implement or manage the items mentioned.)

- Unless the system administrator is specifically authorized by a manager, no one's print jobs will receive priority over other print jobs. (The system administrator can move any print job to the top of the queue, but doing so invariably causes friction among users who have to wait longer because someone received special treatment.)
- Printers are to be used only for projects related to the organization (be it a school, business, or other type of organization). The policy may include a statement about using printers for personal work after certain hours with payment of a fee (such as five cents per page) to a fund for replenishing paper and toner.
- Each user's printer usage will be tracked and recorded. Any user found to be printing an excessive number of sheets may need to explain why. A specific page limit per week or month is sometimes stated. This feature, called print accounting, is especially useful for schools, where many users are using the printer.
- No one should alter the printer settings without instructions from the system administrator. This includes changing the default paper size or default paper tray, changing fonts, adding printer memory, and so forth.

Some of these statements might seem draconian or simply inappropriate within your organization. You can decide which will be helpful in maintaining the printers and the system administrators in good working order.

The `lpd` daemon can record the number of pages printed by each user on the system—a process known as print accounting. To implement print accounting, you need to add a single entry to the `printcap` file. (The following section explains how to add this and other entries.) Once the `lpd` daemon begins print accounting, you can use a simple command-line utility to print reports showing how many pages each user has printed to each printer.

For most printers, page-based accounting is appropriate. If you use plotters on your Linux system, print accounting measures the number of lineal feet that a user has printed.

Creating a `printcap` Entry

The `printcap` file uses a format that is different from any of the Linux configuration files described so far in this book. You can edit this file in any text editor, or you can use one of the graphical utilities provided with your Linux distribution. The general format consists of a printer name (which is actually the print queue name), followed by a series of two-character option codes that apply to that printer. Each option is separated from the next by two colons. Many options are followed by specific parameters. For example, the parameter `lp=/dev/lp0` specifies that the line printer device (the `lp` option code) is `/dev/lp0`. The `lpd` daemon is

designed to read the options defining a printer on a single line. To make the file easier for a person to read, however, each option code is often placed on a separate line of the file, with a backslash indicating that the line break should be ignored when processing the file. This creates a readable format in which many lines are effectively combined into a single line by `lpd`. A very simple `printcap` file from Red Hat Linux is shown below. The Printer Tool (described later) stores some information in the comment lines of the `printcap` file, which makes hand editing this file on Red Hat Linux somewhat problematic.

```
# /etc/printcap
#
# Please don't edit this file directly unless you know what you
are doing!
# Be warned that the control-panel printtool requires a very
strict format!
# Look at the printcap(5) man page for more info.
#
# This file can be edited with the printtool in the
control-panel.

##PRINTTOOL3## LOCAL cdj500 300x300 letter {} DeskJet500 3 {}
lp:\
      :sd=/var/spool/lpd/lp:\
      :mx#0:\
      :sh:\
      :lp=/dev/lp0:\
      :if=/var/spool/lpd/lp/filter:
```

Table 13-1 lists the two-letter codes you are most likely to use when creating a `printcap` file. Be careful about the formatting of this file if you create it manually in a text editor (rather than using a graphical utility as described in the next section). If a single colon, backslash, or equal sign is misplaced, printing may not function correctly. To see a list of all available option codes, use the command `man 5 printcap` to view the `printcap` man page.

Table 13-1 Useful Option Codes Within `/etc/printcap`

Option code	Description	Example
`af`	The print accounting file.	`af=/var/adm/lp_acct/hplj_acct`
`bq`	Define a remote system that should receive the print job, but filter the print job before sending it to the remote system.	`bq=ps@192.168.100.67`
`ff`	The form feed (start a new page) character.	`ff='\f'`
`if`	The print filter to use when printing to this printer.	`if=/var/spool/lpd/hplj/filter`
`lp`	The device name to open for printing (the first parallel port is shown in the example).	`lp=/dev/lp0`

Table 13-1 Useful Option Codes Within `/etc/printcap` (continued)

Option code	Description	Example
`mx`	The maximum file size for print jobs submitted to this printer (in 1024-byte blocks). The first example indicates a maximum print job size of about 2 MB; the second example (with #0) indicates that no maximum print job size is enforced.	`mx=2000` `mx#0`
`pc`	Price per page (or foot for plotters) to use for print accounting, in hundreds of a cent. The example shows 15 cents per page.	`pc=1500`
`pl`	Page length, in lines.	`pl=66`
`pw`	Page width, in characters.	`pw=80`
`rg`	Restricted group; only members of the group named may submit print jobs to this printer.	`rg=eng`

`rm`	Hostname of a remote print server to which print jobs submitted using this printer name should be sent. (See the section "Remote Printing" later in this chapter.)	`rm=ps.xyzcorp.com`
		or
		`rm=192.168.100.67`

`rp`	Remote printer; the queue name to use on the remote print server (defined by `rm`).	`rp=hplj5`
`rs`	Restrict users trying to print from other machines to those with an account (or the same name) on this machine.	`rs`
`sb`	Use a short banner page of only one line.	`sb`
`sc`	Suppress multiple copies of the same print job even if they are specified in the `lpr` command submitting the job.	`sc`
`sd`	The spool directory where submitted print jobs will be stored by `lpd`.	`sd=/var/spool/lpd/hplj`
`sh`	Suppress printing of a banner or header page containing the username, filename, and other print job information. The banner page is helpful when many users use the same printer, but it uses a lot of paper.	`sh`

After creating an appropriate `printcap` entry for the printer you want to use, you should create the print spool directory and the print accounting files (if you have included the `af` option code). The print accounting file, described in detail later in the chapter, stores information about how many pages each user has printed. Assuming you have created a `printcap` entry for a printer named `hplj`, the following commands would create a print spool directory for that printer and set its ownership and permissions appropriately. The following commands assume you are logged in as `root` and that you have specified the directory shown here using the `sd` (spool directory) option code.

```
mkdir /var/spool/lpd/hplj
chown daemon.daemon /var/spool/lpd/hplj
chmod 755 /var/spool/lpd/hplj
```

If you are using print accounting and have indicated a print accounting file with the `af` option code, you can create the print accounting file using similar commands. You can create the file in any location that you choose, though it's best to place it somewhere in the `/var` directory to follow Linux conventions for file placement. You should create the accounting files before starting the printer with the `lpc` command or a graphical utility.

```
touch /var/adm/lp_acct/ hplj_acct
chown daemon.daemon /var/adm/lp_acct /hplj_acct
chmod 755 /var/adm/lp_acct/hplj_acct
```

Using the `lpc` Utility

The **`lpc`** utility is the printer control utility. (The name `lpc` stands for *line printer control*.) This command-line utility lets you control the actions of the `lpd` daemon, specifying how print jobs are accepted and processed. Because many Linux system administrators use one of the graphical utilities described in the next section, they don't regularly need to use the `lpc` utility. But `lpc` provides functionality that is not available in most of the graphical utilities. For example you can:

- Prevent new print jobs from being accepted into a print queue
- Prevent print jobs from being sent to a printer
- Cancel a print job that is currently being printed
- See the status of any printer (whether it is enabled, whether the corresponding print queue contains any print jobs, and so forth)

You can include an `lpc` command as a parameter on the command line, like this:

```
lpc status
```

You can also use `lpc` in an interactive mode in which you can enter multiple commands. To begin using `lpc` in interactive mode, enter the utility name without any parameters:

```
lpc
```

You then see a prompt like this:

```
lpc>
```

Table 13-2 lists the commands that you can use at the `lpc` prompt or as parameters to the `lpc` command (if you prefer not to use the interactive mode for some tasks).

Table 13-2 `lpc` Commands

Command	Description	Example (noninteractive mode)
`help`	Display a list of all `lpc` commands.	`lpc help`
`status`	Display status of the `lpd` printer daemon and the print queue indicated (or the default if none is indicated).	`lpc status`
`abort`	Cancel the print job currently being printed. (Use the `start` command to restart printing.)	`lpc abort`
`stop`	Stop sending print jobs to the printer after the current print job has finished printing.	`lpc stop`
`start`	Start sending print jobs to the printer (used after `abort` or `stop`).	`lpc start`
`disable`	Prevent users from submitting new print jobs.	`lpc disable`
`enable`	Allow users to submit new print jobs.	`lpc enable`
`down`	Stop sending print jobs to the printer and prevent new print jobs from being submitted (equivalent to using `stop` and `disable`).	`lpc down`
`up`	Begin sending print jobs to the printer and allow new print jobs to be submitted (equivalent to using `start` and `enable`).	`lpc up`
`clean`	Remove all temporary files used by `lpd` for the default (or named) printer.	`lpc clean`
`exit` or `quit`	Exit the `lpc` program.	Used only in interactive mode.
`restart`	Attempt to restart the `lpd` printer daemon (equivalent to using the command `/etc/rc.d/init.d/lpd restart`).	`lpc restart`
`topq`	Move a print job to the top of the named print queue. This command requires a printer name (where the print job will be placed) and the print job number (which you can obtain using the `lpq` command).	`lpc topq hplj 16`

For each of the commands shown, `lpc` acts on the default printer (the first one listed in the `printcap` file) unless you specify a printer name. For example, if the first printer in `printcap` is `lp`, using this command will bring down the `lp` printer:

```
lpc down
```

If the `printcap` file contains a definition of another printer named `hplj`, you can use this command to bring down the `hplj` printer:

```
lpc down hplj
```

When using any of the `lpc` commands (except `topq`), you can use the designation `all` to refer to all printers. For example, to completely shut down printing, so that no print jobs can be submitted and all printing stops after the current print jobs are finished, use this command:

```
lpc down all
```

The printing commands you will learn later in this chapter use a `-P` parameter before the printer name, without a following space, for example, `lpq -Phplj`. The `lpc` command does not use the `-P` parameter; you simply add the printer name as an additional parameter.

Figure 13-4 illustrates how some of the key `lpc` commands affect different parts of the printing process.

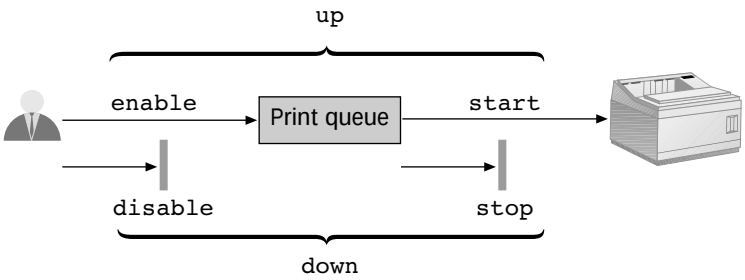

Figure 13-4 Effect of various `lpc` commands on print processing

Using Graphical Tools to Set Up Printing

After reviewing the long list of `printcap` entries in Table 13-1 and `lpc` commands in Table 13-2, you'll be glad to know that many system administrators rely on the graphical tools described in this section to set up and administer printing on their Linux systems. Several different tools are available, though the tools at your disposal depend on your Linux distribution. For example, only Red Hat Linux users are likely to have the LinuxConf and Printer Tool utilities; only SuSE Linux users will have the YAST utility, and only Caldera OpenLinux users will have COAS and LISA. This section describes how to use some of these popular utilities.

You'll note as you explore any of these utilities that they do not provide all of the options listed in Table 13-1 or in Table 13-2. As with much of Linux, if you need to use the full functionality of the Linux printing system, you will need to edit the configuration files yourself. However, many networks have very simple printing requirements, in which case the capabilities of the graphical utilities should be sufficient.

In this section you learn about utilities for setting up the `printcap` file so that users can begin to print files. Later in this chapter, in the section "Managing Printing," you learn about graphical utilities used to manage a print queue.

In Red Hat Linux, you can use the Printer Tool to set up multiple print queues in your `printcap` file. Though powerful, the Printer Tool program is not integrated with any of Red Hat's other system administration utilities. (In particular, you will not find the Printer Tool in the LinuxConf program.) This utility is located on the `AnotherLevel` menu's submenu, under `Administration`. You can also start this utility by entering `printtool` (all lowercase) in a graphical terminal emulator window. If you have started the Red Hat Control Panel (by selecting `Control Panel` from the `System` submenu), you can click on the printer icon to launch the Printer Tool program. Figure 13-5 shows the Red Hat Control Panel with the printer icon in the middle of the panel.

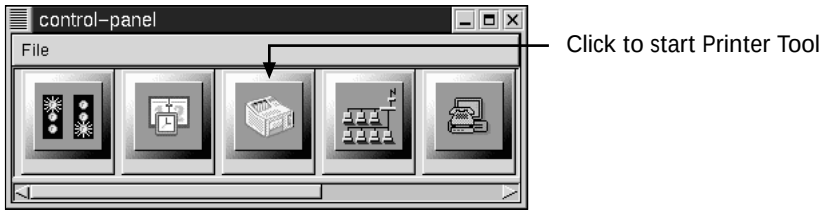

Figure 13-5 The Red Hat Linux Control Panel

When you start Printer Tool, you see a main window with a list of all the printers that are configured on the system. To use the Printer Tool program, choose the `Add` button at the bottom of the main window. In the `Add Printer` dialog box that appears, select the type of printer you want to define. You can choose from the following four options:

- `Local Printer`: a printer connected to the Linux computer via a parallel or serial cable.
- `Remote UNIX`: a remote server that runs the `lpd` daemon and will accept standard UNIX or Linux print jobs over the network.
- `SMB/Windows 95/NT Printer`: a printer connected to a Windows system on the network. This option does not use `lpd`. It uses the Samba suite, which is not discussed in this book. (See *www.samba.org*.)
- `NetWare Printer (NCP)`: a printer connected to a NetWare server on the network. This option does not use `lpd`. It relies on the NCP protocol, which must be installed on your server. NCP networking is not discussed in this book. To learn more, visit the technical support pages at *www.redhat.com*.

Assuming you choose `Local Printer`, you see an informational dialog box listing the printer ports (parallel ports) detected on your system. Choose `OK` to close this dialog box. The `Edit Local Printer Entry` dialog box shown in Figure 13-6 appears. In this dialog box you define options for the printer. Figure 13-6 shows typical values for a local printer. When you use a graphical tool like Printer Tool, the print spool directory that you name in this dialog box is created for you by the utility.

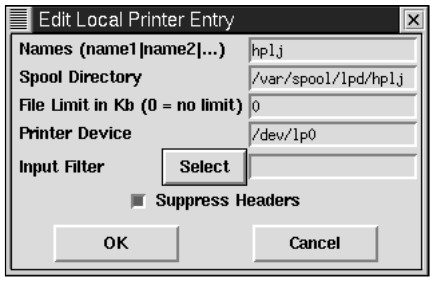

Figure 13-6 Selecting print options in the Red Hat Printer Tool program

As you can see in Figure 13-6, the Printer Tool program allows you to specify relatively few printer options. To add other options, you must edit `printcap` using a text editor. However, you have a powerful tool in the `Configure Filter` dialog box, which you open by clicking the `Select` button next to the `Input Filter` field. Figure 13-7 shows the dialog box, in which you select a printer type and then choose options for that printer. Although this dialog box doesn't include all printer models or options, it allows you to automatically create complex print filters for many popular printers. The options displayed for the printer type that you select reflect the current capabilities of the filtering programs (such as `gs` and `nenscript`), not the full capabilities of the printer model.

The `Configure Filter` dialog box uses the term driver because that term is more familiar to most users than print filter. But in fact, the Printer Tool program creates a print filter, not a driver as in Windows.

The Printer Tool program includes a few menu options that let you print a text page to the selected printer, restart the `lpd` daemon if you are having problems with the printing system, or reload the `printcap` file if you have changed it in a text editor. When you define a printer, the `printcap` file is immediately and automatically updated.

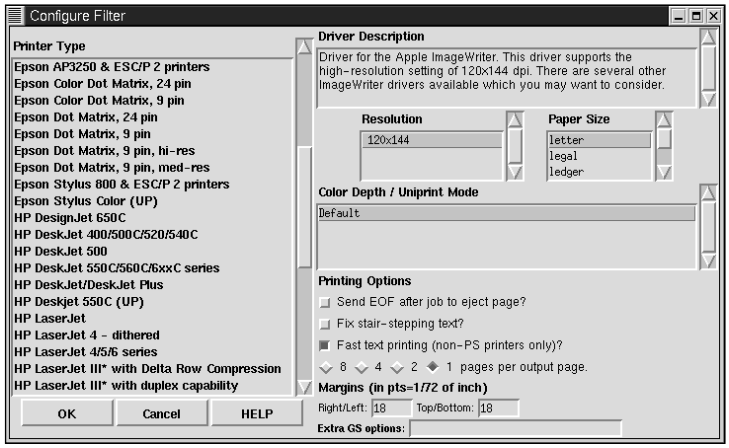

Figure 13-7 The `Configure Filter` dialog box in the Printer Tool

After creating a new printer entry in the `printcap` file, you don't need to restart the `lpd` daemon. The `printcap` file is always read each time a print job is submitted, so the latest information in the file will always be used.

Caldera OpenLinux and SuSE Linux both include configuration utilities that let you define a `printcap` entry in text mode using a menu-style interface. The SuSE utility is called YAST. The Caldera utility is called LISA.

To use the LISA utility, you select `System Configuration`, then `Hardware Configuration`, then `Configure Printer`. Alternatively, you can start the LISA utility at the printer configuration screen using this command: `lisa --printer`. Figure 13-8 shows the LISA utility. After selecting a printer model, you respond to a few questions based on the model that you selected. The `printcap` file and spool directory are created for you.

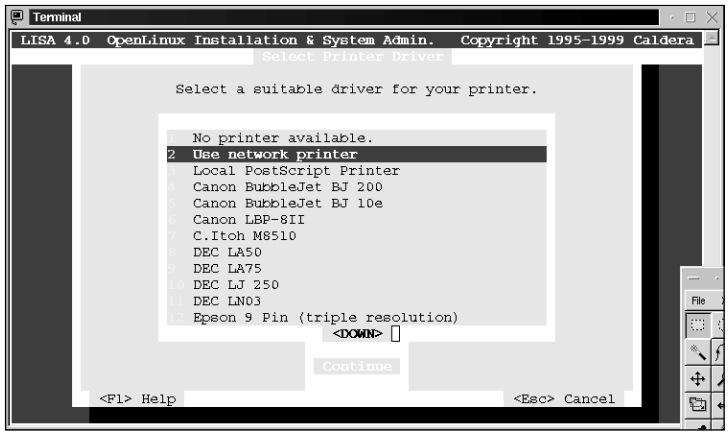

Figure 13-8 The LISA utility in Caldera OpenLinux

Caldera OpenLinux also includes a graphical printer configuration tool as part of its COAS administration system. On the main menu of the KDE Desktop, choose `COAS`, then choose `Peripherals`, then choose `Printer`. When the Printer configuration window appears, choose `Add` from the `Printer` menu. You can then select a printer model from the list and answer other questions about the printer and your preference settings to complete the configuration. The `printcap` file is updated when you confirm (in a pop-up dialog box) that you want your changes saved.

The most well integrated print configuration tool is in the Corel Linux distribution. Although Corel Linux is designed to be used as a desktop system rather than as a Linux server, its printing tools may find their way into other Linux distributions, so you should take time to become familiar with it. To manage printing from the Corel Linux KDE Desktop, open the main menu, choose `Control Center`, and then choose `Printers`. A list of configured printers displays, as in Figure 13-9. Using this dialog box you can do any of the following:

- Configure a local or remote printer using an interface very similar to a Windows wizard.
- Designate a default printer.
- Set advanced options for a printer, such as page size, resolution, and so forth. (You must still edit the `printcap` file to use advanced administration options, however.)
- View and manage the print jobs in any print queue. (Graphical utilities for managing print jobs in other Linux distributions are described in the next section.)

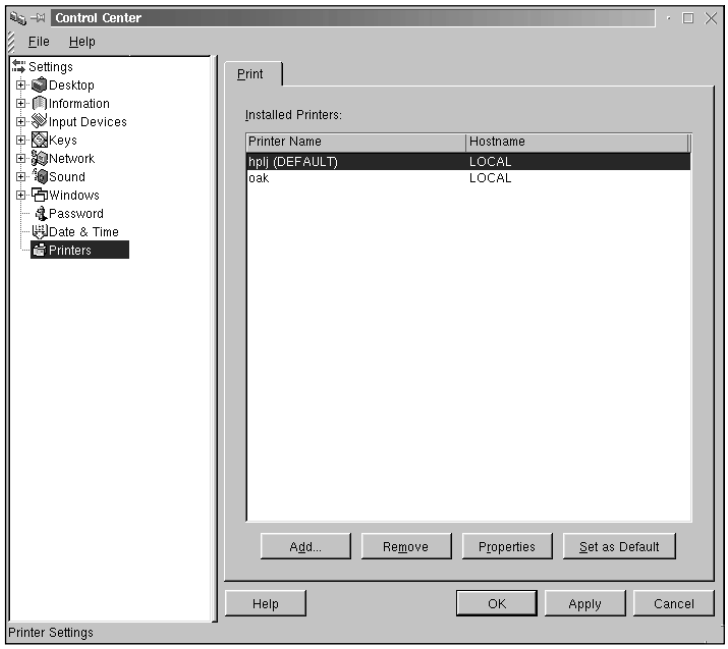

Figure 13-9 Managing printing in Corel Linux

Printing a File

After all of this configuration effort, users can begin to print files on Linux either from the command line or from applications. As mentioned earlier, the command `lpr` prints a file in Linux. The `lpr` command uses the information in the `printcap` file to submit a file to a print queue. From there the `lpd` daemon sends it, in its turn, to the printer. The `lpr` command requires only the name of the file to print. For example, to print a copy of the `printcap` file to the default printer, use this command:

```
lpr /etc/printcap
```

You can also use a pipe symbol to send information to a printer. The `lpr` command reads from the standard input if a filename is not supplied. For example, to sort a file and send the results to the default printer, use this command:

```
sort namelist | lpr
```

One disadvantage of using the `lpr` command in this way (with a pipe symbol rather than a filename) is that the print job information does not contain a filename. Thus you cannot identify what the print job contains after you have submitted it. Still, using the pipe symbol is sometimes useful. The `lpr` command includes many options to control how the print job is handled. Some of the more useful `lpr` options are shown in Table 13-3.

Table 13-3 `lpr` Command Options

Option	Description	Example
`-C`	Set a priority class for the print job. Priorities range from `A` to `Z`. The default is `A`. Print jobs of a lower priority are printed after all print jobs of a higher priority.	`lpr -C D stats.txt`
`-h`	Don't use a banner or header page for this print job, even if `printcap` specifies one.	`lpr -h stats.txt`
`-i`	Indent each line of the print job by this many spaces. This feature is not supported on all printers.	`lpr -i 5 stats.txt`
`-J`	Assign a name to this print job. The name appears on the banner page to help a user identify the print job.	`lpr -J "Weekly statistics" stats.txt`
`-#`	Specify the number of copies of the file to be printed. Note that there is no space between the `#` option and the numeric value.	`lpr -#3 stats.txt`
`-m`	Send an e-mail to the named user account if an error occurs during printing. If the user account is located on the same system, only the user's account name is needed, as in this example.	`lpr -m nwells stats.txt`

Table 13-3 `lpr` Command Options (continued)

Option	Description	Example
`-P`	Specify a nondefault printer using a printer name from the `printcap` file. Don't put a space between the `P` and the printer name.	`lpr -Phplj5 stats.txt`
`-w`	Specify the page width in characters.	`lpr -w132 stats.txt`

The `lpr` utility determines which printer to send a print job to using the steps outlined below. For most users, the default printer will simply be the first one listed in the `printcap` file, because none of the other items in this list are set up by default on most Linux systems. You can change that if you prefer. Note that this procedure also applies to the print queue management utilities (such as `lpq`) described in an upcoming section.

1. If the `-P` option is included with the `lpr` command, the printer named after `-P` is used.
2. If the `-P` option is not included, but an environment variable named `PRINTER` exists for the user printing the file, the value of the `PRINTER` variable is used.
3. If a `PRINTER` environment variable is not configured but a default_printer configuration line is included in the `lpd.conf` file (see the following tip), then that printer is used.
4. If none of the configurations in Steps 1 through 3 provide a printer name, the first printer in the `printcap` file is used.

A configuration file named `lpd.conf` is included with some distributions of Linux, usually in the `/etc` directory. The man page contains details about using this file for advanced configuration of the `lpd` daemon. For example, you can change the location where `lpd` looks for the `printcap` file and which network port `lpd` uses to connect to remote print servers. The configuration settings in this file are beyond the scope of this book. To learn more, contact your Linux vendor's technical support department.

Printing from a graphical program is straightforward. In most cases, you choose `Print` from the `File` menu, review the print options, and choose `OK` to print the current file. Because Linux programs don't have the advantage of full-featured printer drivers, the print dialog boxes are generally very simple affairs. You can occasionally select how many copies to print or select which printer to print to (with the list of available printers being drawn from the `printcap` file). Some print dialog boxes let you select which print command to use, with `lpr` or `nenscript` being the default command. Figure 13-10 shows this feature in the Print dialog box of the KDE text editor.

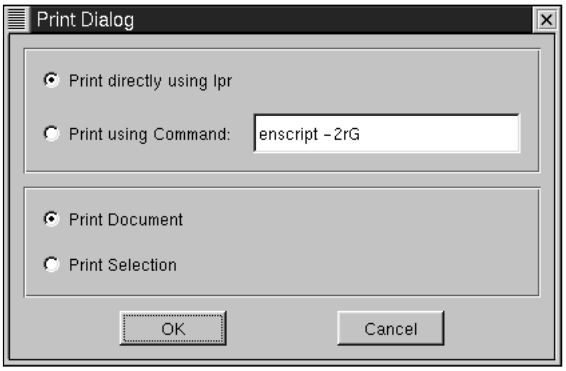

Figure 13-10 Printing a file in the KDE text editor

A few Linux programs such as StarOffice, WordPerfect, and FrameMaker for Linux include more substantial Print dialog boxes, though even these rarely have the depth of printer feature control that Windows users currently enjoy. Figure 13-11 shows the Print dialog box in Corel WordPerfect for Linux. Many additional features are available in this dialog box via the `Select` button in the Current Printer section of the dialog box.

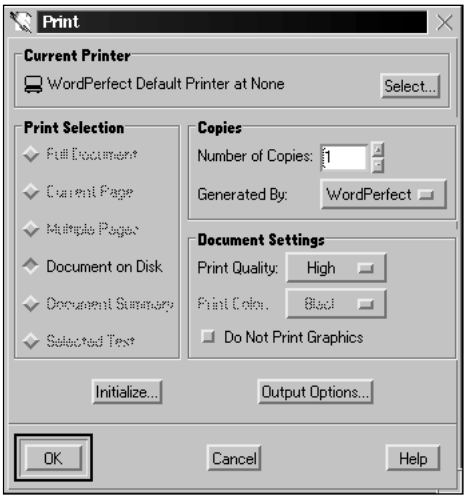

Figure 13-11 The Print dialog box in Corel WordPerfect for Linux

MANAGING PRINTING

Once you have set up a `printcap` file and used the `lpc` program or a graphical utility to enable a printer, managing printing revolves around two tasks:

- Keeping the printers themselves stocked with paper, toner, ribbons, ink, cleared of paper jams, and online. (This book doesn't provide any guidance on these tasks.)
- Tracking print jobs in the print queues.

Linux includes both command-line and graphical utilities for tracking print jobs. Any user on the system can use these utilities, though their functionality is limited if you are not working as `root` because you can only change the status of print jobs that you have submitted.

Tracking Print Jobs

To view the print jobs in the default print queue, enter the command `lpq`. The **`lpq`** utility lists each of the print jobs in a print queue with the following information:

- Status of the print job (such as active, ready, or error)
- Owner (the user who submitted the print job)
- Class of the print job (this is normally `A`, unless the `-C` option was used with the `lpr` command)
- Job number assigned by the `lpd` print server
- Size of the file in bytes (characters)
- Time that the print job was submitted

Sample output from the `lpq` command is shown here:

```
lp is ready and printing
Rank    Owner      Job    Files           Total Size
active  nwells     308    stats.txt       2241 bytes
1st     cynthia    312    index.html      9617 bytes
2nd     alexv      323    userguide.ps    434898 bytes
```

If you are working on a server where the print queue contains many documents, you may either want to use the `-s` option, to display a shorter format for each print job, or add a job ID to the `lpq` command. The job ID refers to the print job number assigned by `lpd` and displayed under the `Job` field in the sample `lpq` output above. For example, to see information about all print jobs submitted by user `nwells`, enter this command:

```
lpq nwells
```

Or if you have previously used `lpq` to identify the job number of a large print job you submitted, use the number to query the status of that job, as in the following command. (The print job number would be different in each case, of course.)

```
lpq 572
```

As with the `lpr` command, you can include a `-P` option with the `lpq` command to view print jobs submitted to different print queues. For example, to view the print jobs in the `hplj` print queue, use this command:

```
lpq -Phplj
```

Once you know the number of a print job, you can remove it from the queue or, if you are logged in as `root`, move the print job to another print queue or to the top of a print queue so that it is printed next. The **`lprm`** command deletes a print job from a queue. It requires a job ID parameter, which can be a print job number or a username. For example, if you have determined (using `lpq`) that a print job with a number of 491 should be deleted before it is printed, use this command to remove it from the `hplj` print queue:

```
lprm -Phplj 491
```

You can remove all print jobs submitted by a user by referring to the user's name, for example:

```
lprm -Phplj nwells
```

You can also remove all of the print jobs in a queue by using the parameter `-`:

```
lprm -Phplj -
```

Obviously, you cannot remove another user's print jobs with a username parameter or the `-` parameter unless you are logged in as `root`.

Creating Print Accounting Reports

When you include the `af` option in the `printcap` file, each print job creates a print accounting entry in the file specified by the `af` option. You can use the **`pac`** (print accounting) command to display reports based on that accounting data.

Although a price per page or foot can be set using the `pc` option in the `printcap` file, the `-p` option of the `pac` command is a more convenient method of seeing a total cost of pages printed by a particular user. As an example of using the `pac` command with the `-p` option to specify a price, consider the command `pac -p0.15 –Phplj`, which will list all the accounting information for printer `hplj` and assign a value of 15 cents per page. Notice the unusual format: neither `-p` nor `-P` has a space after it. Because users will often be printing from multiple locations to a centralized print server (see "Remote Printing," later in this chapter), the `-m` option is commonly used to combine all of a user's print jobs from multiple servers into a single listing. For example, adding this option to the previous example

would yield: `pac -m -p0.15 -Phplj`. Sample output from a `pac` command is shown here, with some lines before the total removed for the sake of brevity:

```
Login          pages/feet      runs                price
nwells         23              4           USD     3.45
toma           490             17          USD     73.5
cynthia        134             6           USD     20.10
dalbis         88              2           USD     13.2
alexv          829             44          USD     124.35
...
total          4319            213         USD     647.85
```

Because the `pac` command uses all of the information stored in the print accounting file that you specified with the `af` option in `printcap`, you may want to set up a script (perhaps a cron job) that periodically rotates the accounting files. Such a script could create a report of print activity, delete or archive the print accounting files for all printers, and then create new, empty files (using the `touch`, `chown`, and `chmod` commands discussed earlier in this chapter).

If you want to see print accounting information for a single user rather than wait for a weekly report to be generated by the script you created, add the user's name to the `pac` command. For example, to see the report for `nwells`, use this command: `pac -m -p0.15 -Phplj nwells`. When a username is not specified, `pac` creates a report line for each user who has printed anything to the specified printer.

Using Graphical Print Management Utilities

In addition to the `lpq` and `lprm` commands, a few graphical utilities are available to help you manage print jobs. Two examples are the `klpq` program (part of the KDE Desktop) and the printer management dialog boxes in Corel Linux. To start the `klpq` program from the KDE main menu, choose `Printer Queue` from the `System` menu. The main window is shown in Figure 13-12.

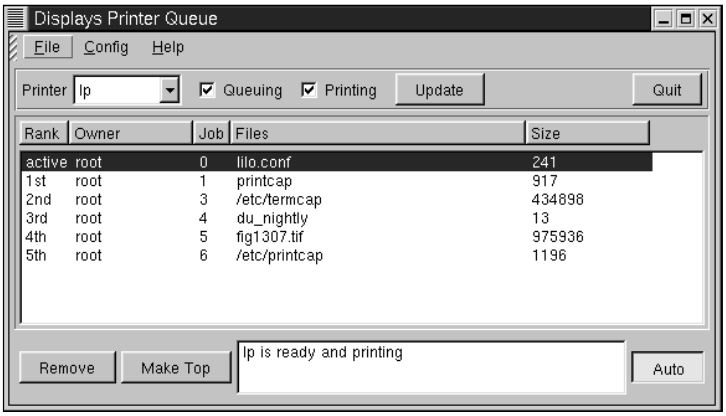

Figure 13-12 Available printers and queued print jobs listed in the KDE Printer Queue Utility

The main window lists all the jobs for the print queue selected in the Printer list box. You can choose which print queue's jobs are listed by selecting a print queue from the list. Each printer in the `printcap` file is included in the list. The Queuing check box, when checked, allows users to submit new jobs to the print queue (when unchecked, new jobs cannot be sent). You check or uncheck the Printing check box to control the `lpd` daemon, which sends print jobs to a printer device. These are equivalent to the `lpc` commands `enable/disable` and `start/stop`, respectively. Remember to click the `Update` button after changing one of the check boxes.

If you are logged in as `root`, you can click on any print job listed and either remove it or make it the next job to be printed. The list of print jobs is updated every few seconds. You can set the update frequency using the `Auto Update` item on the `Config` menu.

Corel Linux takes a slightly different approach from `klpq`, integrating the print queue management into the same dialog box where new printers are defined. To see how this works in Corel Linux, open the Control Center, and choose the `Printers` item; the list of printers

defined on the system (shown earlier in Figure 13-9) displays. Click a printer to select it, and then click the `Properties` button. A dialog box appears that contains a Job Queue tab (shown in Figure 13-13). On this tab you can see the owner, filename, and size for each print job. If you are the `root` user or the owner of a print job, you can use the `Remove` button to delete the print job from the queue.

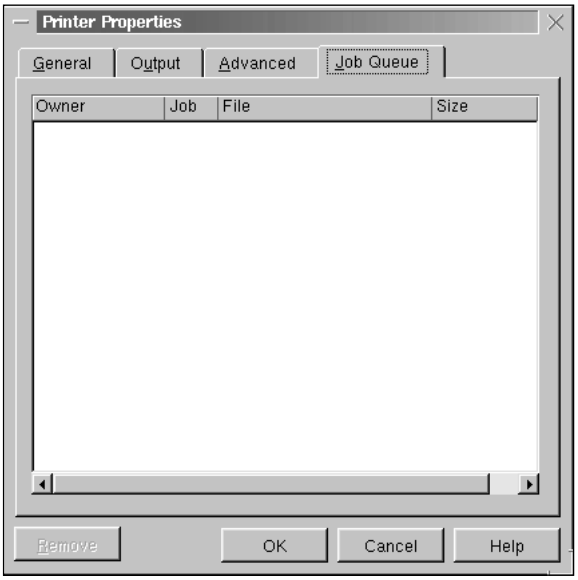

Figure 13-13 Print queue management in Corel Linux

REMOTE PRINTING

Though it's not apparent from what has been said so far, the `lpd` daemon is a network service much like other services that you're more familiar with: FTP servers, Web servers, and e-mail servers. Because security is less of a concern generally when printing files, little configuration is required to connect the `lpd` daemon running on one Linux-based computer to the `lpd` daemon running on another Linux-based computer and transfer a print job for remote printing. In fact, any UNIX system that uses the `lpd` daemon can accept print jobs from a Linux system, and vice versa.

Security is less of a concern with printing, but it may still be something you need to keep in mind. The `/etc/lpd.perms` file contains permissions for users who want to use the `lpd` daemon on a Linux system. This file is not used often, but you can use it to restrict printing activity based on usernames, hostnames, and other factors. To learn more about this feature, contact your Linux vendor's technical support department.

Other systems, such as NetWare and Windows NT, can use special `lpd` daemon software to share printers with UNIX and Linux systems. This software can be difficult to configure, however. You may prefer to use the facilities described below that let you configure native NetWare or Windows printing on Linux.

Printing to Remote Linux and UNIX Systems

Linux and UNIX systems that use the standard printing mechanism described in this chapter can share printers very easily. To print a file from one Linux system to a printer connected to another Linux or UNIX system, simply ask the system administrator of the other system for the print queue name and hostname (or IP address) of the computer. Add these to the `printcap` file as the `rp` and `rm` options, respectively, or use one of the utilities described earlier to define a new print queue for the remote printer. A simple `printcap` file that prints to a remote printer is shown here:

```
lp:\
      :sd=/var/spool/lpd/lp:\
      :mx#0:\
      :sh:\
      :rm=ps.xmission.com:\
      :rp=lexmark:\
      :if=/var/spool/lpd/lp/filter:
```

Figure 13-14 shows the process that `lpd` uses to print to a remote printer. This figure illustrates several important points, some of which are limitations in the Linux printing architecture that you should be aware of.

- The print job is not passed through a filter until it reaches the `lpd` program on the remote system. This allows the administrator of the remote system to control what data is sent to the printer at that location, but it may limit what you can do from your Linux system because your programs don't know how the remote printer is configured.
- Once a print job has been submitted to a remote system, the local `lpd` daemon has nothing more to do with it. You cannot query the remote print queue or learn more about the print job without first logging in to the remote system (which requires that you have an account on that system).
- Unless the default settings have been changed in the `printcap` or `lpd.perms` file, any user on the same network can connect to a printer and submit print jobs. This could be a problem in a large organization or one that is concerned about print usage.

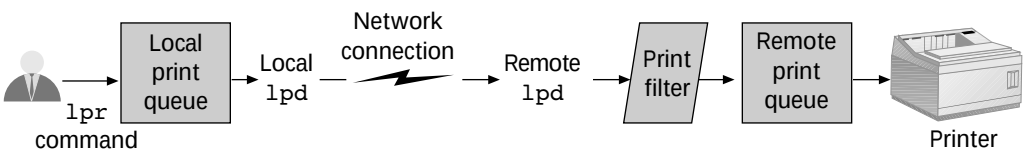

Figure 13-14 Processing a remote print job

Leaving print job filtering to be done on the remote server makes sense in most cases, but it causes a problem when the printer is attached directly to the network. This is a common situation. The printer is considered remote because it is not connected to a serial or parallel port on the Linux computer. Network printers in many companies have their own IP addresses, which makes it possible to send data directly to the printer without connecting it first to a computer's parallel or serial port. Because Linux applications do not preprocess a file based on a printer driver before submitting it for printing, the print filter must do this job. A network printer, however, does not have the ability to run a Linux-style print filter. To solve this problem, a bounce queue command is used.

A **bounce queue (`bq`)** is an option code in the `printcap` file that causes print jobs to be processed by the print filter before being sent to a remote printer. For example, if the `printcap` file includes the following lines, each print job will be processed by the filter, then sent to the print queue named `lexmark2` on the server 192.168.100.43:

```
lp:\
      :sd=/var/spool/lpd/lp:\
      :mx#0:\
      :sh:\
      :bq=lexmark2@192.168.100.43:\
      :if=/var/spool/lpd/lp/filter:
```

Note that no `rm` or `rp` lines are used here, because the `bq` option contains the remote printer information. In addition, no `lp` line is required, because no local printer port will be used to print files. However, as when using the `rm` and `rp` options to specify a standard remote `lpd`-based server, you should include the `sd` option to specify a print spool directory so that `lpd` has a location to store files between the time they are submitted and when they are processed.

Printing to Non-Linux Systems

When you configure a `printcap` file using utilities such as the Printer Tool in Red Hat Linux, you may see references to defining remote printers located on Windows or NetWare systems. The fact that several Linux vendors have integrated multiplatform printer configuration in this way is very helpful for system administrators. But the `lpd` daemon and the print management utilities that you have learned about in this chapter do not have anything to do with printing on these other platforms.

In order to print to a printer that is connected to a Windows-based computer (and not running a Windows version of `lpd`), you must install and configure the Samba suite of applications on your Linux system. This suite of applications provides SMB protocol support to Linux, so that files and printers can be shared between Linux and Windows systems. If you have the username and password required by the Windows security configuration, you can then use the `smbprint` command to send a file to a printer connected to a Windows computer. The Samba suite is included with virtually all Linux distributions, but Samba configuration is far beyond the scope of this chapter.

In order to use a printer that is connected to a NetWare-based computer, you must install and configure the `ncpfs` package or a similar NetWare client package developed by Caldera Systems under license from Novell. The `ncpfs` package is included with many Linux distributions, but it does not support Novell Directory Services—it operates only in NetWare's bindery mode. However, if you have an account on a NetWare server, you can use the `nprint` command to submit files to a NetWare print queue from Linux. Similarly, the NetWare client that Caldera Systems provides includes a utility called `nwprint` that allows you to print to any NetWare printer on your network for which you have access privileges. For more information, visit *www.calderasystems.com*.

CHAPTER SUMMARY

- Printing in Linux is based on the BSD UNIX system. It includes a print server daemon called `lpd` that processes print jobs, the `lpr` program used to submit print jobs, and several other utilities that manage print configuration and print queue management.
- Numerous graphical utilities are available to set up a `printcap` print queue configuration file. These are normally used instead of creating a `printcap` file in a text editor. The `printcap` file contains two-letter codes that define how `lpd` and `lpr` interact with each printer.
- The `root` user on a Linux system can use `lpq`, `lprm`, and `lpc` to manage printing activity and print queue contents. Regular users can manage their own print jobs using `lpq` and `lprm`, or they can use a number of graphical utilities designed for that purpose.
- The Linux printing system is designed for easy sharing of networked printers using one or two options in the `printcap` file. Printing to other systems such as Windows NT and Novell NetWare is also possible using additional Linux software.

KEY TERMS

block special file — A type of file (normally located in `/dev`) referring to a physical device that transfers data in blocks of characters, such as a hard disk drive.

bounce queue (`bq`) — An option code within the `printcap` file that causes print jobs to be processed by the local print filter before being sent to a remote printer.

character special file — A filename referring to a physical device (such as a serial port) that transfers data in single characters.

driver — A software program that provides abstract services for a hardware component, such as opening files or reading character input.

`lpc` — The Linux printer control utility. It stands for *line printer control*.

`lpd` — The line printer daemon, which sends files prepared by `lpr` to the physical printer device or remote print server.

`lpq` — A utility that lists each print job in a print queue and includes information such as the owner and size of each job.

`lpr` — A command that prepares files to be sent to a physical printer device, effectively "printing" files for Linux users.

`lprm` — A command that deletes a print job from a print queue.

magic filter — A print filter program that automatically processes a file into the correct output format based on the file's type.

`pac` — A command that displays reports based on print accounting data. It stands for *print accounting*.

page description language — A special set of codes that determine the graphics elements, text font, and everything else about how information appears on a printed page.

print filter — A script that contains instructions for formatting documents using the page description language required by a specific printer. The print filter is used by the `lpr` program to prepare files to be sent to a physical printer.

print job — A file submitted for printing via the `lpr` command.

print queue — A subdirectory where files are stored to wait for the `lpd` daemon to retrieve them one by one and send them to the printer. Also called a print spool directory.

print spool directory — *See* print queue.

`printcap` — The printer definition file used by the Linux printing system. This file specifies how and where files to be printed are stored and processed by `lpr` and `lpd`. It stands for *printer capture*.

printer policy — A brief document that describes how print resources can be used and how the management of the printers will be conducted within an organization.

REVIEW QUESTIONS

1. Name four components of the BSD UNIX printing architecture used by Linux.
2. A print queue is usually named for:
   a. The print accounting file
   b. A physical printer device
   c. The printer control program
   d. LPRng
3. Print queue definitions are stored in the _____ file.
   a. `/var/spool/lpd/lp`
   b. `/var/adm/acct`
   c. `/etc/printcap`
   d. `/etc/lpd.conf`
4. The relationship of print queue to physical printer can be one to many or many to one. True or False?

5. Print filters are used to:
   a. Convert graphics formats before printing images
   b. Prepare documents in a printer-specific format
   c. Remove unprintable characters from documents
   d. Compress print job files before transfer to a remote print server
6. Describe at least two purposes of having a printer policy in place.
7. Multiple options in the `printcap` file are separated by:
   a. An equal sign
   b. Two colons
   c. A carriage return/new line
   d. A tab
8. Define the meaning of the following option codes as used within the `printcap` file: `sh`, `mx`, `sd`, `if`, and `af`.
9. The `lpc` utility *should* be used to bring up a new printer *unless*:
   a. The graphical utility being used has taken care of this step.
   b. The `lpd` daemon has not been first restarted.
   c. The `printcap` file contains a `bq` option.
   d. The user is logged in as `root`.
10. Explain the difference between the `lpc` commands `up`, `enable`, and `start`.
11. Name four graphical tools that you can use to help set up or manage printing in Linux.
12. Graphical tools are *not* helpful for setting up printing when:
    a. You have only `root` account access.
    b. You require advanced printing security or accounting features.
    c. You are printing to a remote or non-Linux printer.
    d. You are using a standard Linux distribution.
13. You can use the `lpr` command to print a file using standard command-line redirection operators. True or False?
14. Explain the use of the `PRINTER` environment variable.
15. The _____ utility displays the owner, size, and submission time for print jobs.
    a. `lpq`
    b. `lprm`
    c. `lpc`
    d. `lpd`

16. The command `lprm -a kate` would do the following:
    a. Remove all print jobs from a printer called `kate`.
    b. Remove print jobs submitted by user `kate` from all print queues on the system.
    c. Remove the print job that will print the file `kate`.
    d. Remove the accounting files for printer `kate`.
17. The `pac` utility relies on files defined by the _____ option in the `printcap` file.
    a. `ac`
    b. `pa`
    c. `af`
    d. `pc`
18. Graphical utilities are sometimes useful because they combine the functionality of numerous command-line utilities (such as `lpc`, `lpq`, and `lprm`) into a single graphical interface. True or False?
19. Describe two limitations of the remote printing architecture implemented by Linux using `lpd`.
20. The `lpd` daemon is a network service similar to a Web server. True or False?
21. The _____ feature would be useful for printing to a departmental printer that was connected directly to the Ethernet network.
    a. bounce queue
    b. magic filter
    c. print accounting
    d. `ncpfs nprint`
22. Printing utilities such as the Red Hat Printer Tool are able to implement printing to Windows-based computers by relying on what additional software package?
23. Linux does not include a complete set of _____ but instead relies on print filters and programs such as `gs` and `enscript`.
    a. printer drivers
    b. magic filters
    c. `lpd` permissions
    d. remote service packs
24. Contrast the interactive and command-line modes of `lpc`.
25. Linux print utilities commonly determine which print queue to act on by checking:
    a. The `-P` option, then the `PRINTER` environment variable, then the `printcap` file
    b. The `PRINTER` environment variable, then the `-P` option included with the command
    c. The order of print queues in the `printcap` file
    d. The systemwide printer configuration in `/etc/sysconfig`

HANDS-ON PROJECTS

Project 13-1

In this activity you create a `printcap` file with a printer definition, and then you use the `vi` editor to add an option to that printer definition. To complete this activity you should have a working Linux installation with `root` access. You don't need access to a printer.

1. Log in to Linux as the `root` user.
2. If you are using Red Hat Linux, start the graphical environment, and then follow the steps below. (If you are not using Red Hat Linux, skip to Step 3.)
   a. Open a terminal emulator window.
   b. Enter the command **`printtool`** to start the Printer Tool program.
   c. Choose the **`Add`** button.
   d. Select **`Local Printer`** and choose **`OK`**.
   e. Review the printer port autodetection notice, and choose **`OK`** to close the dialog box. The `Edit Local Printer Entry` dialog box appears.
   f. In the Names field enter the printer name **`hplj`**. (You can choose another if you prefer, but use it throughout this project wherever you see `hplj`.)
   g. In the Printer Device field enter **`/dev/lp0`**.
   h. Choose the **`Select`** button. In the `Configure Filter` dialog box select a printer model, explore the available options, and then choose **`OK`** to close the dialog box.
   i. Choose **`OK`** to finish the new print queue configuration.
   j. Choose **`PrintTool`**, then choose **`Quit`** to close the utility.
3. If you are not using Red Hat Linux, use another type of graphical utility to create a basic `printcap` file. You can choose any printer port or input filter that you like; you will not print anything to the printer in this project. You can also use a text editor such as `vi` to create or edit the `printcap` file. Enter this text in the `/etc/printcap` file:

```
hplj:\
        :sd=/var/spool/lpd/hplj:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:
```

4. Enter **`ls -l /var/spool/lpd/hplj`** to check whether a spool directory was created for the printer. If you see a message stating that there is no such file or directory, create the spool directory by entering the following:

```
mkdir /var/spool/lpd/hplj
chown daemon.daemon /var/spool/lpd/hplj
chmod 755 /var/spool/lpd/hplj
```

5. Open the `printcap` file in a `vi` text editor (or another, if you prefer). Within the entry for the `hplj` printer that you have defined, insert the following line to restrict access to this printer to only those users who are members of the `eng` group:

   ```
   :rg=eng:\
   ```

6. Save your changes to `printcap` and exit the `vi` text editor.

Project 13-2

In this activity you use the `lpc` command to control the print queue that you created in Project 13-1. You also set up the group that is allowed to print (`eng`) and print a file to the newly created printer. To complete this activity you need a working Linux installation with `root` access, and you should have completed Project 13-1. You don't need access to a printer.

1. Log in to Linux as the `root` user.
2. Enter **`lpc status hplj`** to check the status of the `hplj` printer that you created in Project 13-1.
3. Enter **`lpc stop hplj`** to stop the `lpd` daemon from attempting to send any print jobs from the `hplj` print queue to a physical printer.
4. Enter **`groupadd eng`** to create the `eng` group (which is allowed to print to `hplj`). (On some Linux distributions you may be prompted to include a group ID number with the `groupadd` command.)
5. Enter **`usermod -G eng`** ***`username`*** (replacing *username* with your username) to add your regular user account to the `eng` group. (Be sure to use a capital *G* in the command.)
6. Change from `root` account access to your regular user account using the `su` command with your username (for example: `su - nwells`).
7. Enter **`export PRINTER=hplj`** to designate the printer assigned to the print queue named `hplj` as the default printer that you want to use. (Note that this change will only remain in effect until you exit the current `su` command; you would need to place this command in a start-up script such as `~/.bashrc` to make it permanent.)
8. Enter **`lpr /etc/lilo.conf`** to print a small file to the `hplj` printer. The `-P` option isn't needed because of the `PRINTER` variable that you just set. You can choose a different file than the one shown here if you prefer.
9. Enter **`lpq`** with no parameters to see the print job that you just submitted listed in the print queue.
10. Enter **`exit`** to return to the `root` account.
11. Enter **`lpc disable hplj`** to prevent more print jobs from being sent to the `hplj` print queue.
12. Change to your regular user account again using the command you used in Step 6.
13. Repeat the command you used in Step 8. What difference do you see? Use the `lpq` command. What do you see?

Project 13-3

In this activity you manage a print queue using command-line utilities. You can experiment with the graphical utilities that come with your Linux distribution if you prefer, but this activity relies on the `lpq`, `lpc`, and `lprm` commands. To complete this activity you should have a working Linux installation with `root` access. This project assumes you have completed Projects 13-1 and 13-2.

1. Log in to Linux as `root`.
2. Enter the commands **`lpc enable hplj`** and **`lpc stop hplj`** to allow print jobs to be placed in the `hplj` print queue but prevent `lpd` from trying to print the files.
3. Change to your regular user account using the `su` command.
4. Enter **`lpr -Phplj /etc/printcap`** to print a file. (Choose a different file if you prefer.)
5. Enter **`lpr -Phplj /etc/hosts`** to print a second file. (Choose any file if you prefer, but this file should be different from the one you printed in Step 4 so that you can identify which was first and which was second.)
6. Enter **`lpq -Phplj`** to list the print jobs in the `hplj` queue. Note the job number of the print job for the file `/etc/hosts` (or whatever the second file that you printed might be).
7. Enter **`exit`** to change back to the `root` account.
8. Use the `lpc` command with the print job number that you noted in Step 6. For instance, if the print job is number 415, enter `lpc topq hplj 415`.
9. Enter **`lpq -Phplj`**. What is different about the print queue contents?
10. Enter **`lprm -Phplj 415`**, substituting the number you noted in Step 6 for `415`.
11. Enter **`lpq -Phplj`** again. What has changed in the print queue?

CASE PROJECTS

The University Computer Lab

1. You have just obtained a coveted job as a system administrator at the university's computer lab. This job gives you experience with many technologies, but also plenty of time to do your homework between crises in the lab. The lab contains several Linux servers, many types of workstations where users work, and a bank of six laser printers. The task that takes up the largest part of your time is troubleshooting printing problems that users report when their print jobs don't come out on the printers. Describe some of the questions that you could routinely ask these users to help discover the problems they are having. As you begin this job, what problems would you anticipate having with numerous new users printing on a large computer lab network?

2. You discover over time that the software (print server) side of things is going fairly smoothly, but the printers themselves seem to be breaking down a lot under the heavy use. Assuming you can't charge the students for computer or printer usage, how could you use the features of Linux printing to help justify to your supervisor the expense of upgrading the printing equipment?
3. You occasionally discover a very large print job (over 300 pages) that appears to have been left on the printer when you finish your shift in the computer lab. But when you return the next day, someone has always returned to pick up the unclaimed papers. The print accounting reports don't seem to show any single user in the lab printing a large number of pages. You are suspicious that someone in another department of the university is sending large print jobs to the lab printers and picking them up later. Describe some of the steps you could take in the Linux printing configuration to watch for or prevent this unauthorized printing.